Transformation of a closed 3D triangular mesh to a quadrilateral mesh based on feature edges

Ryuji Miyazaki[†] **and** **Koichi Harada**[††],

† Hiroshima International University, Higashihiroshima, Japan.
†† Hiroshima University, Higashihiroshima, Japan.

Summary

In this paper, we propose a method of applying the Q-Morph to the closed three-dimensional triangular mesh. The Q-Morph is the method in order to transform the triangular mesh to the quadrilateral mesh. The Q-morph begins quadrilateral transformation from the initial front edges which are usually the boundary of the 2D domain. Also, the Q-Morph can be applied to the 3D domain. But, the closed 3D triangular mesh does not contain any boundary edges which can be the initial front edges. Because of this reason, the Q-Morph cannot be applied directly to the closed 3D triangular mesh.

The initial front edges must be the path of the line segments of which each line segment appears as the edge of the mesh. Furthermore, the shape of converted quadrilateral heavily depends on the initial front edges. So, the initial front edge should be generated by taking the change of the shape into account for effective quadrilateral transformation.

Our proposal is to extract feature edges from input triangular mesh based on the semantic mesh partitioning. And, we modify extracted feature edges so that the extracted edges may adapt to the initial front. We use existing mesh partitioning algorithm for feature edge extraction. In general, extracted feature edges may be jaggy. We smooth extracted jaggy feature edges and modify the triangular mesh so that the smoothed feature edges may appear as the mesh edge in order to create the initial front of the Q-Morph.

Key words:

Quadrilateral mesh, Q-morph, Mesh model, Re-meshing, Feature edge.

1. Introduction

Triangular mesh models are widely used for representing 3D shape in various fields such as computer graphics (CG), finite element method (FEM), and so on. However, a quadrilateral mesh is in demand because it is preferable to a triangular mesh in some types of FEM analysis. A quadrilateral mesh also attracts attention in CG research as the method of parameterization or the subdivision scheme.

Many methods for creating a quadrilateral mesh have been proposed [1]–[8]. The Q-Morph [1], one of these methods, transforms the triangular mesh to the quadrilateral mesh. Recently, many methods of hexahedral mesh generation are studied for solid model generation [9]–[11]. However, the need of the quadrilateral mesh remains in the FEM. The Q-Morph method is widely used as understood from the adoption of it for the commercial FEM software ANSYS [12].

In [1], it is mentioned that the Q-Morph can be applied to the 3D triangulated domain while it is proposed originally for 2D triangulated domain. However, the Q-Morph needs the initial front in order to start quadrilateral transformation. Usually, the boundary of the triangulated domain that is transformed to the quadrilateral mesh is used as the initial front. Because of this reason, for a closed 3D triangular mesh that does not have any boundary, it is impossible to apply the Q-Morph directly. In practice, Owen et al. applied the Q-Morph to the 3D CAD model that is separated into some regions in advance [1]. Thus, some boundaries must be defined to apply the Q-morph to the closed 3D triangular mesh.

The shape of the quadrilateral that is created by the Q-Morph depends heavily on the initial front. So, we must define the initial front appropriately for applying the Q-Morph to the closed 3D mesh consisting of various shapes. It is considered that some feature edges of the mesh can be used as the initial front. The traditional feature edge extraction is based on the local change of the surface. For example, the feature edge is extracted from the mesh by thresholding the angle between triangles that share the edge. Such a method extracts many redundant rough edges. The initial front for the Q-morph must be the closed boundary of the domain that is transformed to the quadrilateral mesh. Usually, it is not guaranteed that extracted feature edges compose the closed boundary suitable to the initial front. So, it is difficult that the traditional method extracts appropriate initial front for the Q-Morph.

Marinov et al. proposed a re-meshing method that uses the boundary of the semantically partitioned regions [8]. Such a boundary is aligned along the sharp feature of the input shape while it is not based on the local shape information. It is considered that the boundary of the partitioned region can be used as the initial front. However, the boundary that is extracted by this method is usually jaggy. Such a jaggy boundary is not suitable for using as the initial front for the Q-morph.

In this paper, we propose a method for generating the initial front for the Q-Morph from the closed 3D triangular mesh. The initial front that is generated by our method consists of the closed path of the line segment that appears

Manuscript received May 5, 2009
Manuscript revised May 20, 2009

as the mesh edge. Furthermore, each line segment is connected smoothly. Such a line segment path can be used for creating the quadrilaterals adapted to sharp features of the input mesh.

This paper is composed as follows. Related works are described in the section two. The creation of the initial front edges from the 3D triangular mesh is explained in the section three. In the section four, we mention how to apply the Q-Morph to the 3D triangular mesh. The implementation result is shown in the section five. Finally, we conclude our paper in the section six.

2. Related Works

Some quadrilateral or quad-dominant mesh creation methods have been proposed for FEM. Shimada, et al. proposed a method to create the quadrilateral mesh by packing squares to a surface [3]. They obtain desired position of vertices by packing squares to target surface according to the given vector field. Then, a triangular mesh is created from the obtained vertices. This triangular mesh is converted to a quadrilateral mesh based on directionality of the given vector field.

Blacker et al. proposed the "Paving" quadrilateral meshing method [2]. Owen et al. also proposed the "Q-Morph" to create the quadrilateral or quad-dominant mesh by converting a triangular mesh [1]. These methods create quadrilaterals based on front edges. The front edge is the boundary of the domain that is already transformed to quadrilateral and the domain that is not transformed yet. During the transformation, the front edges are advanced for the region of which quadrilaterals are not created yet. Because of this procedure, these methods are called advancing front method. Alexander et al. improved the Q-Morph in order to create high quality quadrilaterals [5]. Quad-dominant re-meshing is also studied in the CG field. Alliez et al. [4] and Marinov et al. [6] proposed the quad-dominant re-meshing method by drawing streamlines on the surface of the triangular mesh. Streamlines are drawn along the direction of the principal curvature of the surface. Then, quadrilaterals are created by intersections of streamlines. Because the direction of the principal curvature cannot be distinguished at a flat, sphere, or umbilical point, they adopt the extra procedure to improve the curvature direction of such a point by the trusted direction. Dong et al. generate the vector field on the mesh surface by defining the harmonic scalar field over the mesh vertices [7]. This vector field can be made similar to the principal curvature direction by setting appropriate constraints.

Extracting feature vertices and edges from a 3D triangular mesh is a basic technique for detecting features from a 3D mesh. Such a feature provides us important shape information for effective mesh processing such as feature-preserving simplification. Many methods have been proposed [13]–[15]. However, these methods are based on the local shape information.

On the other hand, Cohen-Steiner et al. proposed the variational shape approximation (VSA) method to partition the input mesh based on several error metrics [16]. When the input mesh is partitioned by their $L^{2.1}$ metric, the boundary of regions is aligned to the direction of which the change of the surface is small. Marinov et al. used the 3D cubic curve approximation of the boundary of the region for quad-dominant re-meshing [8].

Our proposal is similar to the method of [8]. However, the approximated boundary curve is not the curve on the mesh and such a curve does not appear as edges of the mesh. Our proposal is to generate the smooth path of the mesh edges from such a boundary for the Q-Morph.

3. Getting the initial front edges from a closed 3D triangular mesh

The Q-Morph starts quadrilateral transformation from the initial front edges. Usually, the boundary of the region that must be transformed to the quadrilateral mesh is used as the initial front edges. That is, creating the initial front edges from a 3D triangular mesh is to extract the closed path of line segments from the input mesh. In addition, every line segment must be the edge of the mesh. The quality of quadrilaterals created by the Q-Morph depends on the initial front edges. Due to this property, the line segments should be connected smoothly.

Our proposal is composed of the following three steps.

1. Partitioning a 3D triangular mesh into semantic regions by the VSA method.

2. Smoothing the region boundary so that the boundary is represented by line segments on the surface of the input mesh.

3. Splitting triangles in order that every line segment appears as the edge of the mesh according to the smoothed region boundary.

3.1 Mesh segmentation by the VSA

The VSA partitions a mesh model into several regions based on the error metric. The VSA defines the proxy consisting of a point and a normal vector. A mesh model is partitioned into the regions approximated by the proxy. The error of partitioned region is defined with respect to its proxy by the error metric. Then, a mesh model is partitioned so that the error of the regions is minimized. The error metric $L^{2.1}$ defined in [16] is based on the

difference of the normal vectors. So, partitioning the mesh by the $L^{2.1}$ metric means that each region is as flat as possible.

Let t_i be a triangle and let n_i be a normal vector of t_i. And, let $P_i = (X_i, N_i)$ be a proxy consisting of a point X_i and a normal N_i. The $L^{2.1}$ error of t_i with respect to P_i is calculated as follows:

$$E(t_i, P_i) = \| n_i - N_i \|^2 | t_i |,$$

where, $| t_i |$ is the area of t_i. Then, the error of the partitioned region R_i with respect to its proxy P_i is defined as follows:

$$E(R_i, P_i) = \sum_{t_j \in R_i} E(t_j, P_i,).$$

The VSA clusters triangles in order to partition the input mesh. The clustering process is carried out by growing the regions from the seed triangles according to the error. In practice, we define some initial seeds randomly for the initial clustering. After clustering, we define the proxy of the clusters by using the area-weighted average of the triangle's normal vector in each cluster. Then, the clustering is improved by using the proxy and new seed triangle that is located at the center of the region. Additional seeds are used for improving quality of clustering. For example, if maximum error of the region exceeds a threshold, an extra seed is added in this region. Then re-clustering is carried out.

A sample partitioning result is shown in Fig. 1. This example started from two seeds that were selected randomly. We add an extra seed in each region if the maximum angle between the triangle and the proxy exceeds $\pi/3$. The obtained partition shows that the threshold value suits the input shape very well. However, the boundary of each region is usually serrate. Such a boundary is not desirable for creating quadrilaterals.

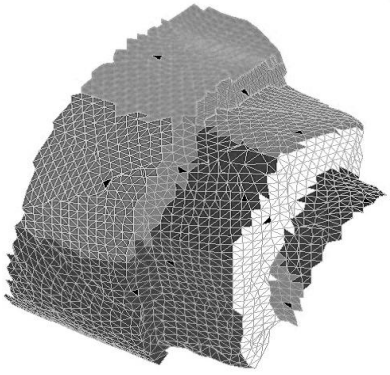

Fig. 1 An example of partitioning a 3D triangular mesh by the VSA. Black triangles show the seed triangle of each region.

3.2 Smoothing the region boundaries

We have obtained the boundary of the regions in the section 3.1. What we need in this subsection is the boundary that consists of the path of the line segments connected smoothly. The method for drawing the path of the line segments on the surface of the mesh is studied as the geodesic path problem. The Dijkstra algorithm is the most famous algorithm to find the shortest path between two points. The Dijkstra algorithm was improved in various ways to draw the optimal geodesic line. The geodesic path by the traditional Dijkstra-like algorithm consists of the edges of the mesh. Thus, obtained geodesic path is usually serrate. Dimas et al. proposed a method to compute approximate geodesics on triangular meshes by straightening serrate geodesics [17]. We utilize this method in order to smooth the serrate boundary.

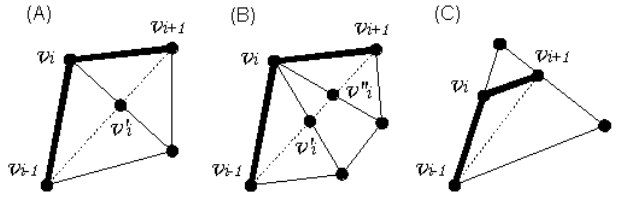

Fig. 2 The modification of vertex position for smoothing the path of the line segment. (A): v_i moves to v'_i. (B): If the line $v_{i-1}v_{i+1}$ intersects two edges, v'_i and v''_i are inserted to the vertex sequence instead of v_i. (C): If the line $v_{i-1}v_{i+1}$ does not intersect to any edges, v_i is removed from the vertex sequence.

The straightening procedure is carried out as follows. The initial geodesic path is expressed by the sequence of the vertices $(v_0, v_1, \cdots v_n)$. The geodesic path is straightened by moving the position of each vertex. The vertex v_i is modified by v'_i. v'_i is the intersection of the mesh edge and the line $v_{i-1}v_{i+1}$, as shown in Fig. 2-(A). If several edges intersect to the line $v_{i-1}v_{i+1}$, extra vertices are inserted to the sequence of the vertices (Fig. 2-(B)). This modifying procedure is repeated until the geodesics are straight enough. During the straightening process, some vertices are removed from the vertex sequence if the line $v_{i-1}v_{i+1}$ does not intersect to any edges (Fig. 2-(C)).

In order to calculate the intersection, triangles of the mesh are flattened locally by rotating the triangle along the edge (Fig. 3). The vertex is classified by the sum μ of its incident angles as:

1. Euclidean if $2\pi - \theta = 0$,
2. Spherical if $2\pi - \theta > 0$, or

3. Hyperbolic if $2\pi - \theta < 0$.

θ is divided by the geodesics to θ_r and θ_l as shown in Fig. 4. Triangles of the side θ_r or θ_l are flattened according to the type of the vertex in order to calculate the intersection.

In [17], the straightening process is repeated until the line segment path is straightened enough. However, we repeat this process only a few times because we need only the smooth line segment, not the straight-line segment. An example of smoothing the boundary is shown in Figs. 5 and 6.

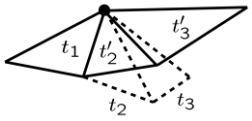

Fig. 3 An example of local triangle flattening. The original triangles are t_1, t_2 and t_3. t_2 and t_3 are rotated to t'_2 and t'_3, respectively.

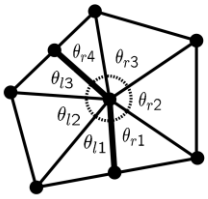

Fig. 4 The sum of incident angles of the vertex. The left sum $\theta_l = \theta_{l1} + \theta_{l2} + \theta_{l3}$. The right sum $\theta_r = \theta_{r1} + \theta_{r2} + \theta_{r3} + \theta_{r4}$.

3.3 Re-meshing according to the smoothed boundaries

Now, we have obtained the smooth line segment path over a 3D triangular mesh. We create the 3D triangular mesh such that the line segment appears as the edge. The re-meshing process is executed by splitting the triangle that the line segment traverses. Our re-meshing process consists of two steps. First, the triangle is split by the line segment. Then, triangulate the remained polygon.

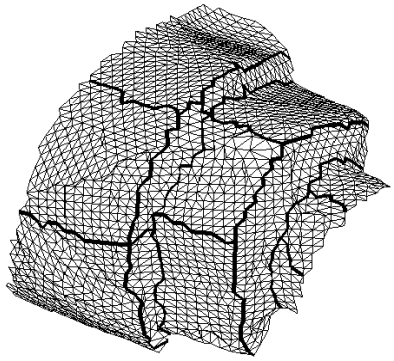

Fig. 5 The boundary of the partitioned regions. The boundary is the path of the mesh edges. But, each boundary is serrate.

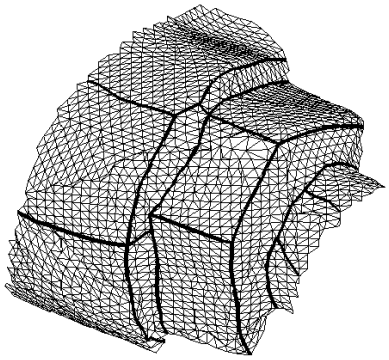

Fig. 6 The result of smoothing the boundary shown in Fig. 5. Each boundary consists of line segments on the surface of the mesh.

We assume that the triangle is represented as a vertex loop. When the line segment traverses the triangle, terminals of the line segment are on the edge of the triangle. Then, we can obtain the looped vertex list consisting of vertices of the triangle and terminals of the line segment.

Triangle splitting is achieved by processing this looped vertex list as follows. If the looped vertex list includes the line segment $LS_{ab} = (v_a, v_b)$, this looped vertex list is split into two looped vertex lists, $(v_a, \cdots, v_b)$ and $(v_b, \cdots v_a)$. For example, two line segments LS_{ab} and LS_{cd} traverse the triangle $t = (v_0, v_1, v_2)$. The looped vertex list LVL is represented as $(v_0, v_a, v_d, v_1, v_c, v_2, v_b)$ (Fig. 7-(A)). When the triangle t is split by the line segment LS_{ab}, the LVL is split into $(v_a, v_d, v_1, v_c, v_2, v_b)$ and (v_b, v_0, v_a). That is, the triangle t is split into the polygon (v_a, v_1, v_2, v_b) that LS_{cd} traverses and the triangle (v_b, v_0, v_a). The splitting process is applied to all line segments in the same manner. In this example, the triangle t is split into two triangles (v_b, v_0, v_a), (v_d, v_1, v_c) and the polygon $(v_c, v_2, v_b, v_a, v_d)$.

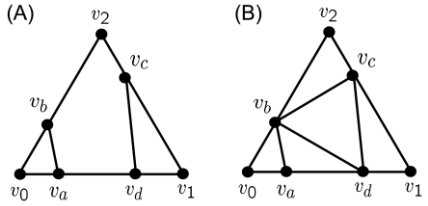

Fig. 7 The procedure for which the line segment appears as the edge of the mesh. (A): The line segments LS_{ab} and LS_{cd} traverse the triangle (v_0, v_1, v_2). (B): The triangle (v_0, v_1, v_2) is split by the line segments LS_{ab} and LS_{cd}. After that, the polygon $(v_c, v_2, v_b, v_a, v_d)$ is triangulated.

After splitting by line segments, we triangulate the polygon that is not triangle. Also, we triangulate the polygon automatically by processing the looped vertex list. If the polygon is represented by the looped vertex list $(v_0, v_1, \cdots, v_n)$, we create triangles from this polygon such that the looped vertex list is split into the triangle (v_0, v_1, v_2) and the polygon $(v_2, \cdots v_n, v_0)$. This triangulating process is repeated until all polygons become triangle. In the case of Fig. 7-(A), the polygon $(v_c, v_2, v_b, v_a, v_d)$ is split into triangles (v_c, v_2, v_b), (v_b, v_a, v_d) and (v_d, v_c, v_b) (Fig. 7-(B)). This process is applied to all triangles that the line segment paths traverse. Finally, the line segment path appears as the mesh edge. Then, we can use the line segment path as the initial front edges for the Q-Morph. An example of splitting triangles is shown in Figs. 8 and 9. Fig. 9 shows that the triangle is split according to the line segment and the line segment appears as the edge of the mesh.

4. Applying the Q-Morph to a 3D triangular mesh

The Q-Morph is an algorithm for transforming the 2D triangular mesh to the quadrilateral mesh. The Q-morph needs to determine the intersection of the vectors during the quadrilateral transformation.

Two vectors may intersect in 3D space if these are on the same plane. So, in order to determine the intersection of two vectors, these are transformed to the vectors that are on the same plane. Then, it can be determined whether two vectors intersect or not. The Q-Morph determines the intersection of the vectors in 3D space by this way.

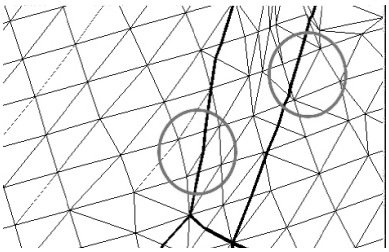

Fig. 8 A part of Fig. 6. The line segment traverses the triangle.

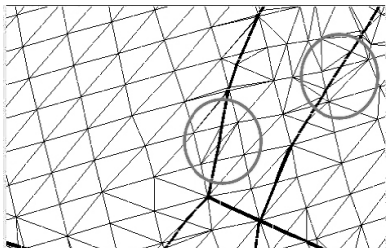

Fig. 9 The result of splitting triangles according to the line segment. The line segments appear as the edge of the mesh.

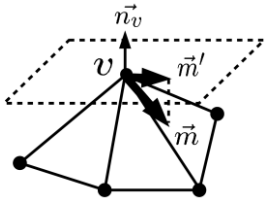

Fig. 10 An example of the movement of the vertex. The vector $\vec{n_v}$ is the normal of the vertex v. The vector $\vec{m}$ show the direction which is calculated by the smoothing process. The vector $\vec{m}'$ is the orthogonal mapping of $\vec{m}$ to the plane of which the normal is $\vec{n_v}$ and passes v. The vertex v moves to the direction $\vec{m}'$.

However, a modification is necessary for applying the Q-Morph to the 3D triangular mesh. It is to eliminate the inverted triangles after the quadrilateral creation. Moving vertices of which the inverted triangle is made up eliminates the inverted triangle. In general, the smoothing process moves vertices so that distribution of vertices becomes uniformly. Uniformly distribution of vertices can avoid inverting of triangles. Various smoothing algorithm can be used for moving the vertex. However, it is known that the smoothing process yields shrinkage of the shape. So, in order to prevent the shrinkage problem, we move the vertex to the direction that is orthogonal to the normal vector of the vertex (Fig. 10).

5. Implementation

In this section, we demonstrate our proposal creating a quadrilateral mesh from a 3D triangular mesh. Fig. 11 shows the 3D triangular mesh model. This is the RockerArm model from the Cyberware Inc.[18]. We use this mesh as the input mesh for our first demonstration. This mesh is simplified by the Qslim software [19] in order to reduce calculation time; our proposal could apply to dense meshes for detailed quadrilateral mesh creation.

Fig. 12 shows the initial front edges that are generated by our proposal. The initial front edges are drawn as thick line. The initial front edges are generated on the boundary between the primitive shapes like a cylindrical or a flat surface. In addition, the initial front edges are aligned along the direction of small surface change.

Fig. 13 shows the result of applying the Q-Morph that starts from the initial front edges of Fig. 12. The initial front edges are shown as the thick edges. In this implementation, we change the procedure of creating the quadrilateral a little in order to preserve the initial front edges. The Q-Morph applies special procedures "seam", "transition seam" and "transition split" in some conditions. The seam and transition seam procedures are necessary for the case of which two adjacent front edges make sharp angle. The transition split is applied in general for the case when the length of adjacent front edges is much different. Because alignment of the front edges is disarranged by these procedures, we do not apply these procedures for the initial front edges. We leave the triangle instead of applying these procedures in order to preserve alignment of the initial edges. From Fig.13, it is shown that quadrilaterals are aligned along the direction of small surface change.

Figs. 14, 15 and 16 show the other demonstration. Fig. 14 is the input mesh model of the RabbitSculpture that is also from the Cyberware Inc.. The input mesh is also simplified by the Qslim. This input mesh consists of the surfaces of gradual bent. Fig. 15 shows the initial front edges generated by our proposal. In this example the initial front edges are also aligned along the direction where the change of the shape is small. Fig. 16 is the result of applying the Q-Morph by using the initial front edges shown in Fig. 15. Quadrilaterals are placed suitably to the direction of the change of the shape even if there is not the sharp feature in the input mesh.

The performance of the implementation is shown in Table 1. The table shows that this result is practical for the pre-processing of the re-meshing.

	Mesh clustering	Boundary smoothing	Triangle splitting
Fig. 13	21.0s	6.3s	0.9s
Fig. 16	19.2s	6.1s	0.8s

Table 1: Calculation time of the examples.

6. Conclusion

We proposed a method to generate the initial front edges for the Q-Morph quadrilateral mesh creation from the 3D triangular mesh that does not have any boundary edges. We implemented the traditional Q-Morph directly in 3D space, in order to show that our proposal can generate adequate edges for the Q-Morph. Moreover, we can consider two improvements.

One is the creation of well-shaped quadrilaterals. The shape of the quadrilateral created by the Q-Morph depends not only on the initial front edges but also the initial triangulation. Desired initial triangulation consists of equilateral triangles whose size is nearly the same. Sophisticated smoothing algorithms like in [20] give such a triangulation. Furthermore, improved Q-Morph, QMV(Q-Morph Variation)[5], can achieve creation of higher quality quadrilateral mesh.

The other is to avoid processing the mesh in 3D space. Implementation of the Q-Morph in 3D space needed some modifications. Especially, undesirable shrinkage problem caused by smoothing should be solved. Recently, some methods for mapping 3D triangular mesh onto 2D domain are proposed in the computer graphics field. Our proposal creates the mesh that consists of semantic regions with smooth boundary. Such a region can be mapped easily onto appropriate 2D domain. We can apply the Q-Morph to a triangulated 2D domain of which the 3D triangular mesh is mapped. Then the 3D quadrilateral mesh can be recovered from the quadrilateral mesh of 2D domain.

In recent years, the re-meshing methods according to the principal directions of the mesh surface have been widely studied. It is known that the mesh of faces aligned along the principal directions can represent the object shape very well. For example, edges of such a mesh are aligned along the principal directions. It yields smooth highlight line over the mesh surface. However, the principal directions cannot always be defined at all point of the surface. The Q-Morph method creates the quadrilateral mesh without taking account of the vector flow such as principal directions. As the future work, we try to create the mesh whose edges aligned along the change of the shape by giving appropriate initial front edges for the Q-Morph.

Acknowledgments

The mesh model of the RockerArm and the RabbitSculpture are open to the public by the Cyberware Inc.. We wish to express our gratitude to the Cyberware Inc..

References

[1] S. Owen, M. Staten, S. Canann, and S. Saigal, "Advancing front quadrilateral meshing using triangle transformations," Proceedings of 7th International Meshing Roundtable, pp.409–428., 1998.
[2] T. Blacker and M. Stephenson, "Paving: A new approach to automated quadrilateral mesh generation," International Journal for Numerical Methods in Engineering, vol.32, no.4, pp.811–847, 1991.
[3] S. Kenji, J. Liao, and T. Itoh, "Quadrilateral meshing with directionality control through the packing of square cells," Proceedings of 7th International Meshing Roundtable, pp.61–76, 1998.
[4] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, and M. Desbrun, "Anisotropic polygonal remeshing," International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2003 Papers, pp.485–493, 2003.
[5] A.V. Skovpen, "Modified algorithm for unstructured quadrilateral meshing," RFNC-VNITF, 2004.
[6] M. Marinov and L. Kobbelt, "Direct anisotropic quad-dominant remeshing," Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04), pp.207–216, 2004.
[7] S. Dong, S. Kircher, and M. Garland, "Harmonic functions for quadrilateral remeshing of arbitrary manifolds," Computer Aided Geometric Design, vol.22, no.5, pp.392–423, 2005.
[8] M. Marinov and L. Kobbelt, "A robust two-step procedure for quad-dominant remeshing," Computer Graphics Forum, vol.25, no.3, pp.537–546, 2006.
[9] N. Chiba, Y. Yamashita, C. Takizawa, and K.Fujishiro, "An automatic hexahedral mesh generation system based on the shape-recognition and boundary-fit method," Proceedings of 5th International Meshing Roundtable, 1996.
[10] S. Owen and S.Sunil, "H-morph: An indirect approach to advancing front hexahedral meshing," International Journal for Numerical Methods in Engineering, vol.1, no.49, pp.289– 312, 2000.
[11] M. Hariya, I. Nishigaki, I. Kataoka, and Y. Hiro, "Automatic hexahedral mesh generation with feature line extraction," Proceedings of 15th International Meshing Roundtable, 2006.
[12] "Ansys. inc." http://www.ansys.com/.
[13] T. Watanabe and H. Chiyokura, "Extraction method of feature edges of a shape modeled by using arbitrary triangular mesh," IEICE TRANSACTIONS on Information and Systems, vol.J83-D-II, no.5, pp.1344–1352, 5 2000.
[14] K. Watanabe and A. Belyaev, "Detection of salient curvature features on polygonal surfaces," Computer Graphics Forum, vol.20, no.3, 2001.
[15] A. Belyaev and E. Anoshkina, "Detection of surface creases in range data," IMA Conference on the Mathematics of Surfaces, pp.50–61, 2005.
[16] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2004 Papers, pp.905–914, 2004.
[17] D. Martinez, L. Velho, and P. Carvalho, "Computing geodesics on triangular meshes," Computers & Graphics Journal, vol.29, no.5, pp.667–675, 2005.
[18] "Cyberware." http://www.cyberware.com/.
[19] M. Garland and P. Heckbert, "Surface simplification using quadric error metrics," Proceedings of SIGGRAPH '97, pp.209–216, 1997. http://graphics.cs.uiuc.edu/» garland/ home.html, retreived on April 2007.
[20] L. Chen, "Mesh smoothing schemes based on optimal delaunay triangulations," Proceedings of 12th International Meshing Roundtable, pp.109–120, 2004.

Ryuji Miyazaki is a lecturer in the Department of Kansei Design at Hiroshima International University. He received the BE and MS in 1996 and 1998, respectively, from Hiroshima University. His current research is mainly in the area of computer graphics. He is especially interested in 3D shape modeling. He is a member of IEICE and Japan Society of Kansei Engineering.

Koichi Harada is a professor of the Graduate School of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and PhD in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of computer graphics. Special interests include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, and IPS of Japan.

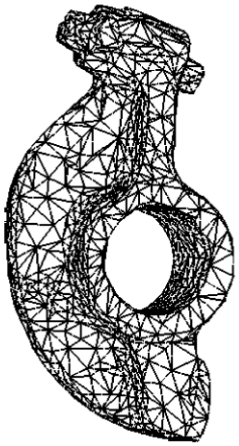

Fig. 11 The input mesh of RockerArm model from the Cyberware Inc. (1678 vertices and 3356 triangles).

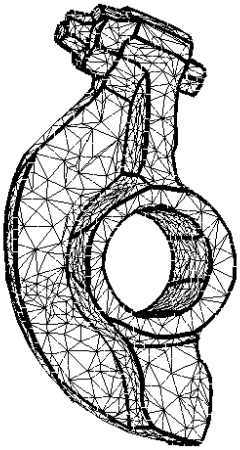

Fig. 12 The result of generating the initial front edges from Fig. 11. The initial front edges are shown as the thick edge.

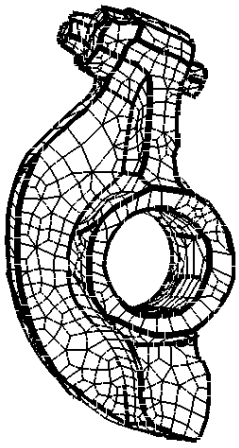

Fig. 13 The result of quadrilateral transformation of Fig. 11 by the Q-Morph. The initial front edges are shown as thick edges.

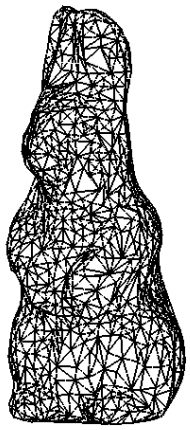

Fig. 14 The input mesh of RabbitSculpture from the Cyberware Inc. (1280 vertices and 2568 triangles).

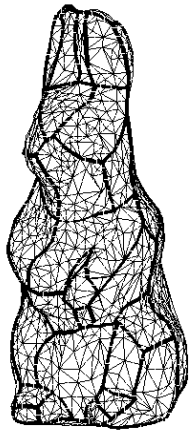

Fig. 15 The initial front edges of Fig. 14 (thick line).

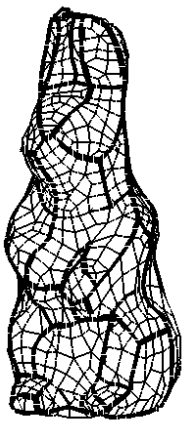

Fig. 16 The quadrilateral mesh which is transformed from Fig. 14. The initial front edges are shown as thick edges.